SUPLEMENTO MENSUAL DEDICADO A LA IMAGEN SINTÉTICA

RENDERMANÍA

Número 9

La Portada

Escaramuzas virtuales

Biblioteca 3D

- Caballeros y zombis para Imagine y POV
- 3dto3d, un conversor para Imagine

Taller Virtual

Books.inc, un nuevo Ipas para POV

pcmania

José Manuel Muñoz

Books.inc: Un Ipas para

La afirmación de que las nuevas sentencias de programación de POV son la novedad más importante de la versión 3 de este raytracer puede parecer excesiva, pero dicha afirmación está siendo confirmada por las nuevas utilidades que están apareciendo. En el número anterior nos asombramos con «Trees» de Sonya Roberts y hoy volveremos a hacerlo con «Books», un "Ipas" de la misma autora para crear filas de libros virtuales.

crear librerías con POV

Hemos decidido denominar "Ipas" a las nuevas utilidades de POV escritas con el lenguaje escénico de este Ray tracer, ya que casi todo el mundo sabe lo que es un Ipas (un módulo de software externo que es llamado por 3D Studio cuando es necesario y que se apoya en las propias funciones del programa). Los Ipas de 3D Studio emplean una interfaz de comunicación con 3D Studio para acceder a las funciones del programa. Esta interfaz viene a ser una "puerta abierta" mediante la cual los programadores pueden ampliar las posibilidades del programa en sentidos muy diversos.

Quien conozca el mundillo de 3D Studio sabrá que se han escrito Ipas para casi cualquier cosa. Pues bien; esto mismo es lo que está empezando a suceder en el universo POV, con la diferencia de que la creación de Ipas para POV es una tarea mucho más sencilla que la equivalente para 3D Studio: cualquier usuario de POV con nociones de programación puede diseñar su propio Ipas. Así, podemos escribir Ipas para crear árboles, campos de meteoritos, edificios, etc.

Seguidamente estudiaremos books.inc. Este Ipas, que genera filas o pilas de libros, puede no ser tan útil o espectacular como otro capaz de generar árboles, pero es un ejemplo perfecto de lo que puede conseguirse aunando imaginación con habilidad de programación. El fichero es, además, relativamente sencillo, y podrá ser comprendido incluso por aquellos povmaníacos que sientan escalofríos ante la palabra "matemáticas".

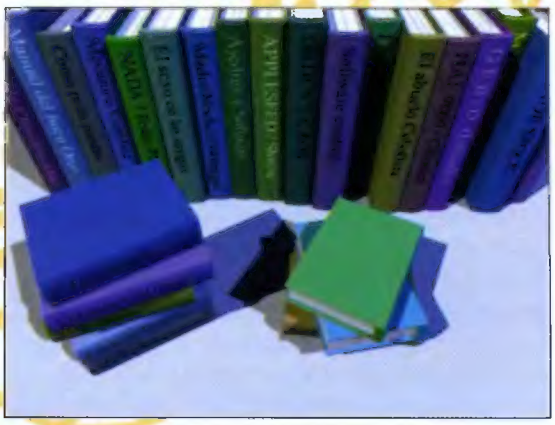
Filas de libros

Books.inc permite al usuario generar filas de libros que pueden ser empleados para dar un acabado más completo a cualquier escena interiorista. Podremos pues emplear este Ipas para llenar estanterías con libros o para crear pilas de volúmenes sobre las típicas mesas de trabajo. Los libros que genera Books no tienen un nivel de detalle demasiado alto, pero, a pesar de esto, las escenas quedarán resultonas, ya que los volúmenes podrán tener distintos grosores, colores y tamaños, y podremos variar su alineación dentro de las filas. Además, de manera similar, podremos hacer que las pilas de libros tengan un aspecto desorganizado, acorde con el que puede verse en las pilas de libros del universo real (esto parece ser una preocupación constante en el trabajo de Sonya Roberts).

Como sucedía en Trees, el usuario controlará el funcionamiento de Books mediante una serie de variables cuyo valor se introducirá desde el fichero escénico, antes de colocar el correspondiente include. El número de estas variables es bastante grande pero, por fortuna, no es preciso que las referenciemos todas. De hecho Sonya ha dado valores por defecto a cada una de estas variables empleando la sentencia ifndef. Veamos un ejemplo:

```
#ifndef (RowLenght) #declare RowLength=2 #end
```

En esta línea, situada al principio del archivo books.inc, se estipula que si la variable RowLenght no ha sido inicializada antes (desde el fichero escénico del usuario), esta pasará a tener un valor de 2. Este sistema es muy cómodo para el usuario ya que se ahorrará la introducción de valores cada vez que books.inc sea referenciado: si el usuario da un nuevo valor a una variable, este valor sustituirá al que se haya asignado por defecto en el ifndef. Además, el nuevo valor podrá ser alterado más adelante en una nueva invocación a books.inc. De lo contrario quedará como nuevo valor por defecto.



Mayor variación en la colocación de la fila.

Variables de control

Ahora examinemos las variables de control. En primer lugar tenemos a "Rowlenght", que es la longitud de la fila o columna de libros a crear. La altura de la fila se controla con "RowHeight" y la profundidad de la misma se indica con "RowDepth". "StackStyle" indica al Ipas la forma en que van a colocarse los libros. Si el valor es 0 (el dado por defecto) los libros se dispondrán de la manera corriente que podemos observar en cualquier estantería. Puesto que Books asume –al igual que la mayoria de los usuarios de POV– que el plano del suelo es el formado por los ejes X-Y, los libros se dispondrán "de pie" formando una fila a lo largo de este plano. Los libros descansarán sobre Y=0 y sus lomos estarán orientados en la dirección -Z. La fila, que arranca en X=0. se extenderá hasta el punto indicado por "RowLenght".

Otra posibilidad es dar a "StackStyle" el valor 1. En este caso los libros formarán una pila como la que puede verse en ciertas mesas, donde el volumen que reposa directamente sobre la mesa tiene encima una docena o más de tomos. La pila será colocada en la posición <0.0.0> y su altura dependerá del valor dado a "RowLenght". En cuanto a la forma de la pila, ésta se dispondrá de una manera bastante ordenada, ya que Books "sólo" aplicará un desorden máximo de 20 grados (utilizando valores aleatorios) en el eje Y; de esta manera se evitará el que los libros estén excesivamente ordenados sobre la mesa, lo que resultaría irreal. Finalmente podemos dar a "StackStyle" el valor 2, con el cual se generará otra pila como la anterior, pero mucho más desordenada, ya que ahora el rango de giro de los libros oscila entre los 0 y los 360 grados en el eje Y.

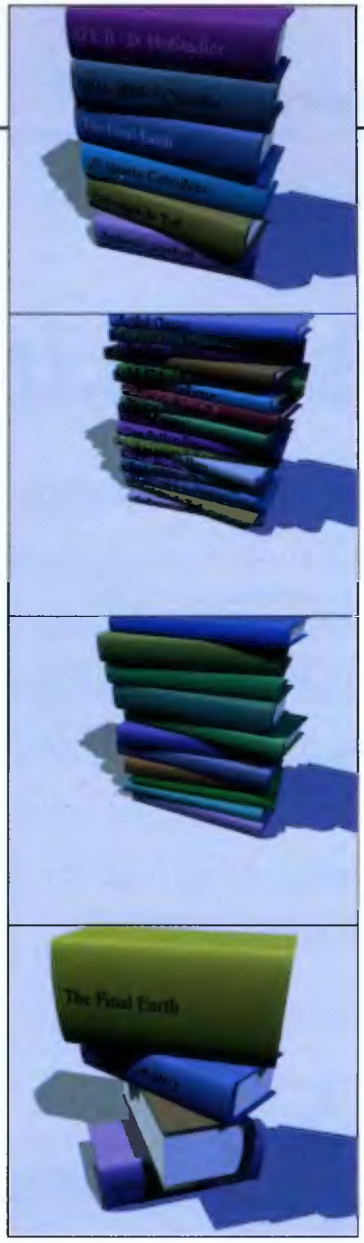
Sin variación de color y generando títulos al azar.

Veamos ahora otras variables. "NumBooks" controla el número total de libros de la fila o pila. En realidad este número es aproximado ya que puede ocurrir que, a consecuencia de las variables que controlan las diferencias en el grosor de los libros, haya libros de más o de menos. En cualquier caso el grosor medio de los libros se calculará dividiendo la longitud dada a la fila en "RowLenght" por el valor de "NumBooks". Hay que tener cuidado con los valores dados a estas variables u obtendremos unos libros ridículamente gruesos o muy finos, y lo mismo se aplica a las variables que controlan la altura de los tomos ("RowHeight") o el largo de los mismos ("RowDepth"). Estas proporciones no son difíciles de calcular pero más vale recordar que los valores de todas estas variables se combinan para componer los resultados.

Sigamos con otros detalles. Si todos los libros generados fuesen excesivamente similares en sus dimensiones, el resultado sería poco atractivo. Lo que queremos es una librería con tomos de diferentes tamaños, como sucede en la vida real. Esto se consigue con las variables "VariThick", "VariHeight" y "VariDepth", las cuales controlan respectivamente las variaciones en el grosor, la altura y el largo de los libros (con respecto a las dimensiones medias, por supuesto). Los valores dados serán decimales y es conveniente que no sean demasiado grandes (observe el lector el resultado de los valores por defecto).

Otra variable importante de este mismo tipo es "VariAlign". Esta se emplea para conseguir leves diferencias en la alineación de las filas de libros. El valor 0.1, asignado por defecto, nos dará filas muy ordenadas y valores mayores producirán filas más desordenadas.

De igual manera podemos conseguir más diferencias entre los libros poniendo a "True" la variable "VariColours". Con ello se asignarán colores aleatorios a las cubiertas de los tomos. Si, por el contrario, dejamos esta variable a "false", los colores asignados se elegirán entre una gama de grises (el modo de color es el elegido por defecto). Aquí hay que decir que "true" y "false" son identificadores de POV. Un valor "true" es –como en el lenguaje C– cualquier valor distinto de 0, mientras que "false" es 0. Por alguna razón Sonya empleó "True" en vez de "true", con el resultado de que el parser de POV declara errores en algunos sitios. El autor de este artículo remedió esto añadiendo



Diferentes pruebas. Nótese como el título puede acabar fuera del libro por excesiva longitud o por poco grosor del libro.

#declare True=true, al principio de books.inc (y haciendo lo mismo con false). ¿Un error de última hora?

Por otro lado, existe una variable "semilla", llamada "BookSeed", que se emplea para inicializar el generador de números aleatorios. Con ello podremos conseguir filas o pilas de libros diferentes, cambiando el valor de dicha semilla (si estamos generando una animación mantendremos el mismo valor de semilla en todos los frames para obtener siempre la misma fila en cada uno).

En cuanto a "FillerColour", controla el aspecto del canal del libro. Si el valor de esta variable es 0 obtendremos un blanco plano mientras que un valor de 1 proporcionará tonos grises.

Los objetos "TEXT"

Ya sólo queda por explicar lo que hace la variable "Titles". Si su valor es 0, las cubiertas de los libros serán planas. Si el valor es 1, habrá títulos en inglés en los lomos de los libros. Realmente el pov-usuario no necesita saber más para emplear Books, pero surgen varios interrogantes: ¿Cómo se colocan las letras sobre los libros? ¿Podemos poner nuestros propios títulos?

Veamos: las letras se colocan empleando los objetos Text implementados en POV a partir de la versión 3.0. Antes de esta novedad el usuario estaba obligado a hacer operaciones CSG o a utilizar Heightfields para crear textos. Ahora, sin embargo, podemos emplear esta nueva primitiva para crear objetos-texto a partir de fuentes TrueType. El formato de la nueva sentencia es:

text { ttf "Nombre del fuente" "cadena a imprimir" Profundidad, Separación }

En "Nombre del fuente" podemos especificar alguno de los fuentes ttf que se han incluido como ejemplo en POV (o cualquier otro de que dispongamos). En cuanto a la cadena, normalmente usaremos un string entrecomillado. También resulta posible emplear una variable de cadena, en cuyo caso bastará con incluir el nombre de la misma, sin las comillas. Los parámetros siguientes determinan la profundidad de la letra y el espacio de separación. Por defecto las letras se colocan sobre el plano X-Y, pudiéndose leer el texto desde -Z. La parte frontal del texto queda en Z=0.1 y cada letra tiene de 0.5 a 0.75 unidades de altura. Así pues, "Profundidad" se refiere a las dimensiones del objeto a lo largo del eje Z. En cuanto al siguiente parámetro hay que tener cuidado, ya que si no indicamos un eje, el valor de separación se aplicará como una suma en los tres ejes (cada letra se alejará de la anterior en cada eje). Esto se evita escribiendo cosas como ".3*x", lo cual restringe cada desplazamiento al eje indicado.

Cadenas

En books los títulos se componen aleatoriamente definiendo un string a partir de la suma de otros tres. La variable string obtenida, "BookTitle", será luego empleada para los objetos ttx. Esta suma se lleva a cabo empleando una de las nuevas sentencias de trabajo con strings de la forma:

#declare BookTitle=concat(First, Second, Third)

La función concat realiza la suma de las cadenas contenidas en las variables. Si por ejemplo, sus valores respectivos eran "The ", "Lost " y "World", el resultado será "The Lost World". Las palabras para cada una de las tres variables-sumando se extraen aleatoriamente (empleando rand y switch) de tres listas que podemos hallar en titles.inc. (este método es similar al que se usaba en broma para hacer que los programas creasen poemas). Sin embargo, y por aquello de crear títulos en castellano, hemos creado un fichero aparte donde las cadenas son títulos completos y no se emplea rand (confiemos en que Sonya no se moleste por ello). Hemos incluido ejemplos de uso. Sólo queda advertir que tengáis cuidado al poner vuestros títulos. ya que podéis exceder el espacio para ponerlos (esto puede arreglarse cambiando el scale del objeto text).

Modelos para una batalla

Durante meses los rendermaníacos han enviado material bélico para participar en la batalla entre Orcos y Humanos que estaba prevista para este mes. Catapultas, cañones, transportes, e incluso un par de guerreros y un mago han llegado a Rendermanía para apoyar a uno u otro bando. Y este mes (¡por fin!) ha llegado la hora del encontronazo virtual. Sin embargo, la horda orca no se ha presentado. En su lugar ha acudido al campo de batalla una horda de no-muertos.

Ante todo, quisiéramos disculparnos por haber sustituido a la horda orca por los zombis que podéis ver en estas páginas. Esto ha sido debido a una imperdonable falta de previsión. En el número anterior nos pareció que ya había material suficiente para preparar una buena batallita virtual, pero faltaba lo más importante: ¡los Orcos!

Por qué no hay orcos

Lo cierto es que, como recordarán los lectores, disponíamos de un orco que había sido cedido amablemente por José González Iniesta. Este orco era un magnífico representante de la raza verde: tenía grandes colmillos, unos hombros de metro y medio de ancho, una apariencia que inspiraba terror al enemigo y, cómo no, desprendía un hedor capaz de poner en fuga a cualquier humano. Sin embargo, al final no pudo ser utilizado. ¿Por qué? Pues porque el orco no había sido diseñado expresamente para una batallita virtual como la que se pretendía preparar. No estaba articulado y únicamente poseía una textura para la parte frontal.

Como el modelo no estaba dividido en dos partes, la textura –creada para representar la parte delantera del orco– quedaba aplicada también en la parte

de fantasía medieval

posterior de la criatura verde, con lo cual no resultaba conveniente que el modelo quedara de espaldas a la cámara (¡entonces se apreciaba que no había espaldas, sino dos lados delanteros!). Así, el orco solamente podía ser visto de frente (bastaba con hacerlo girar unos pocos grados para advertir que estaba incompleto). Quizá se pudo haber creado una escena donde los orcos sólo presentasen la parte frontal a la cámara, pero esto hubiese recortado seriamente las posibilidades en el diseño de la portada. Además, los guerreros humanos estaban articulados y podía prepararse cualquier número de posturas para ellos; por tanto resultaba deseable hacer lo propio con los orcos.

De esta manera se llegó a la conclusión de que se necesitaba un orco tan manejable como el lancero humano. Fue entonces cuando, sabiendo que José González tenía en ese momento trabajo suficiente como para ahogarse en él, decidimos hacer nosotros mismos al orco.

Para ello disponía de la magnífica documentación que supone la revista White Dwarf, donde hay docenas de fotografías de miniaturas de orcos. Pronto, no obstante, hubo que rendirse a la evidencia: preparar un orco mínimamente aceptable no es algo que se haga en unas pocas horas. Quien haya visto alguna de las muchas ilustraciones que existen sobre esta raza fantástica sabrá que los orcos no suelen llevar armaduras. La mayor parte de su cuerpo –parecido al humano pero mucho más fornido– queda al descubierto. Era pues preciso crear un cuerpo antropomórfico que exhibiera lo más fielmente posible la brutal musculatura orca. Hacer esto desde cero ya suponía una tarea muy difícil pero, además, también era preciso crear un rostro que pudiese variar de expresión y que fuese fácilmente deformable, lo cual era –como ya imaginará el lector– una empresa aún más laboriosa. Resumiendo, las alternativas posibles eran:

1) Crearlo todo (cuerpo y cabeza) partiendo de cero.

2) Realizar deformaciones de una malla freeware de un cuerpo humano hasta obtener el orco.

3) Utilizar bitmaps para representar detalles inexistente en un modelo 3D básicamente muy sencillo (como es el caso de los modelos de «Quake» o del propio orco de José González).

4) Crear un modelo muy simple que solamente fuese visto a distancia.

5) Hacer otra cosa.

6) Renunciar a la batallita.

Las dos primeras opciones se pusieron en práctica hasta que resultó evidente que los progresos eran demasiado lentos. ¡Sencillamente no había tiempo de hacer un orco decente en los días disponibles! En cuanto a la tercera opción, nuestra habilidad como grafistas 2D no estaba a la altura de las circunstancias. La siguiente opción, la cuarta, fue rechazada sin más, puesto que se deseaba un orco del que sus colegas verdes no se avergonzaran. Así pues, sólo quedaban dos opciones: o bien renunciar al encuentro o bien "ha-

cer otra cosa". Por supuesto también se hubiera podido aplazar el encuentro pero esto hubiera sido defraudar a todos aquellos que deseaban participar en él. La batalla, pues, debía celebrarse a toda costa, aunque faltaba uno de los bandos...

La noche de los muertos vivientes

Afortunadamente somos auténticos devoradores de literatura fantástico-heróica y por esta razón no tuvimos que estrujarnos los sesos hasta niveles críticos para hallar una solución. Los adeptos al género, quienes hayan leído a Howard, Zelazny, Leiber, Moorcok, Tolkien y a tantos otros, se habrán dado cuenta de que tarde o temprano en estas historias siempre aparece una legión de muertos vivientes (de hecho, en «Warhammer» también figura un ejército de no muertos). Esta fue la solución adoptada por razones obvias. En primer lugar diseñar un zombi es muy fácil, únicamente hay que hacerse con algún modelo de esqueleto y modificarlo para nuestros fines. Además el no-muerto, al consistir solamente en huesos, no presenta las complicaciones de modelado típicas de las caras o de los músculos y resulta muy fácil articularlo (todo el mundo sabe dónde están los ejes de giro de un cuerpo humano).

Una vez tomada esta resolución el siguiente paso a tomar fue considerar dónde iba a tratarse el modelo. O sea, dónde iban a crearse las diferentes posturas que, más tarde, en «POV», se emplearían para generar las escenas. Esta decisión no fue difícil.

«Imagine» dispone de un excelente modelador que resultaba adecuado para crear los pocos objetos necesarios. Permite una fácil modificación de los modelos utilizando operaciones que pueden afectar a objetos, caras o puntos y, sobre todo, es ideal para establecer jerarquías de objetos, lo cual es indispensable para preparar posturas.

El guerrero humano

Una vez decidido el uso de «Imagine» se optó por preparar al guerrero humano en primer lugar. Este era el modelo que iba a representar al bando humano. Fue creado en «3D Studio» y era deseable comprobar si iban a presentarse problemas de traducción de formatos antes de empezar con el esqueleto (que también es un modelo de «3D Studio»). Efectivamente hubo problemas.

Para empezar se suponía que tanto «Imagine» como «3D Studio» pueden leer ficheros dxf. Pues bien, mientras que «3D Studio» no presentó problemas para leer los dxf grabados desde «Imagine», este último si los dio para leer los dxf grabados desde «3D Studio». «Imagine» se ponía a leer y a leer para, finalmente, mostrar un eje solitario en pantalla, pero del objeto leído, ni rastro. Por supuesto, uno siempre tiene que dejar una puerta abierta a la admi-

sión de haber cometido un error o alguna omisión catastrófica, pero en este caso, al no haber encontrado opciones de lectura, simplemente se aceptó que esta opción no funcionaba (probablemente lo hará con archivos dxf más antiguos que los que graba «3D Studio»). Entonces se empleó el conversor Wcvt2pov, ya publicado hace tiempo en PCmanía. Wcvt2pov no realiza traducciones directas al formato de «Imagine» pero permite leer ficheros 3ds y grabar dxf. Después de las primeras pruebas se comprobaron dos cosas:

A) «Imagine» leía los dxf grabados por Wcvt2pov (aunque tardaba un tiempo considerable en leer los más largos) pero convertía los modelos leídos en un único objeto.

B) En caso de que se grabase pieza a pieza como dxf un modelo desde Wcvt2pov, «Imagine» daba a los objetos leídos tamaños que no correspondían a sus dimensiones originales. Las piezas, por tanto ya no encajaban entre sí y se hacía preciso reescalarlas y posicionarlas nuevamente desde «Imagine».

Así pues, al no encontrar un conversor directo 3ds-obj, fuimos importando pieza a pieza desde «Imagine». Para reescalar y situar las piezas se importó también un modelo completo del lancero que sirvió de pista para comprobar las dimensiones y la posición de cada objeto. Después de un trabajo mecánico y aburrido, el lancero quedó listo. Fue entonces y sólo entonces cuando, examinando las cartas del foro del lector encontramos la referencia a 3dto3d en la carta de Víctor Tovío Ciércoles. (Nota del autor: estamos seguros de haber oído antes referencias a 3dto3d pero las habíamos olvidado totalmente. Gracias, amigo Víctor).

3dto3d

3dto3d es un conversor de formatos 3d. Permite una perfecta comunicación entre «POV», «3D Studio» e «Imagine», aunque no pueden pasarse escenas de «POV» a «3D Studio» ni a «Imagine». Las traducciones realizadas con 3dto3d presentan algunos problemillas menores pero la alternativa –como habréis comprobado por los párrafos anteriores– es mucho peor.

La versión que este mes publicamos es la remitida por Víctor. Esta versión ya había sido publicada antes en la revista, pero no lo recordamos a tiempo. La versión funciona bajo MS-DOS por el sistema de la línea de comandos. El formato de uso de la utilidad es:

3dto3d fichero.ext /opción /opción

Hemos de indicar a 3dto3d el formato del fichero a traducir con la opción "i". A este carácter seguirá un valor indicando el formato del fichero de entrada. Un valor de 0 (el asignado por defecto) corresponde al formato Raw. El valor 1 es el de los ficheros 3ds («3D

Studio»), el 2 el de los archivos .obj («Imagine») y el 3 el de los ficheros Lwo («Lightwave»). Para especificar el formato del fichero de salida usaremos la opción "o" seguida de un valor numérico. Este valor será de 0 para los ficheros raw puros (sin almacenar los vectores de las normales), de 1 para los raw suavizados (en los que sí se almacenan estas normales), y de 2 para «POV». Merece la pena indicar que la opción 2 graba también un fichero en formato Udo que –según parece– puede ser leído por «Moray».

Otros valores válidos para "o" son el 3 («Polyray»), el 4 (Asc, el formato ascii de «3D Studio»), el 5 (Obt), el 6 (Rpl de «Real 3D»), el 7 (3ds), el 10 (obj de «Imagine»), el 11 (Rwx de «Renderware»), el 12 (Wrl de «Vrml 1.0») y el 13 (Dxf). Los valores 8 y 9 se emplean respectivamente para generar objetos wireframe y blob para «POV», pero no han sido probados aún.

Así pues, por ejemplo, la línea...

3dto3d nombre.3ds /i1 /o10

...generaría un fichero obj para «Imagine» a partir del fichero de «3D Studio» dado como entrada. Esta conversión no parece presentar problemas pero lo contrario sí puede ocasionar algunos contratiempos. A veces, los modelos obj exportados al formato de «3D Studio» no se visualizan correctamente desde «3D Studio» y puede suceder que la mitad de los polígonos no sean visibles. Si queremos solucionar esto desde el mismo «3D Studio» bastará con unificar las normales del objeto. ¡Los polígonos están ahí! únicamente ocurre que el lado visible de las caras (indicado por la normal de cada polígono) apunta en dirección opuesta a la visible. También podemos añadir la opción /u a la línea de órdenes de 3dto3d. Esta orden –según el manual– unifica las normales del objeto, pero aparentemente no funciona. También puede suceder que el modelo se vea facetado desde «3D Studio» aunque originalmente su superficie estuviese suavizada. Para remediar esto desde «3D Studio» bastará con aplicar un comando de suavizado (nota: 3dto3d también ofrece el comando /s para hacer esto pero, nuevamente, parece no tener efecto).

Por el contrario la conversión 3ds - obj no dio problemas en ninguna de las pruebas realizadas y fue así como el modelo del esqueleto se pasó a «Imagine». Después de la conversión, los objetos componente del modelo siguen siendo independientes en «Imagine». No se fusionan en un único objeto. Todos los objetos individuales se atarán a uno de mayor jerarquía (que, es de suponer, 3dto3d elegirá "a dedo"), y todo el modelo podrá ser referenciado pinchando sobre este objeto "padre". Una vez que tenemos el modelo en la pantalla de «Imagine» podremos desagruparlo y establecer las jerarquías necesarias antes de grabarlo.

Preparación de los luchadores

Una vez cargados los dos modelos en el Editor de Detalles de «Imagine», se crearon las relaciones jerárquicas para poder preparar las posturas necesarias para la batalla. Antes, sin

embargo, se estudió el tamaño relativo entre ambos modelos y se realizaron algunos ajustes de poca importancia en los dos. Quizá el más importante sea el realizado en las manos. Aunque crear posturas nuevas de modelos como estos puede ser bastante divertido, colocar la posición de las manos es siempre una tarea molesta. Simplemente hay demasiados objetos y puntos de giro. El usuario pierde siempre un buen rato colocando la posición de cada dedo y a menudo se encuentra con que la disposición obtenida no resulta adecuada y hay que volver a empezar.

Para estos personajes se decidió suprimir el problema de raíz. En los dos, las manos están cerradas, como asiendo algo, y forman un único objeto atado a la muñeca. La idea surgió ojeando una White Dwarf. Al contemplar las fotografías, observamos que prácticamente todas las miniaturas tenían las manos cerradas en un puño, lo cual era lo más natural del mundo ya que siempre asían algo: un escudo, una espada, una lanza... Incluso en los raros casos en que una mano quedaba libre, ésta formaba un puño. Esto deja abierta la posibilidad de que el personaje agarre el mismo arma con las dos manos (como se ha hecho en un par de posturas del zombi) o de que se pueda colocar otro arma fácilmente. Una vez decidido esto se preparó la postura para la mano esquelética y se unió (join) en un único objeto. La mano del humano fue algo más complicada ya que la que se preparó originalmente fue sustituida por otra recortada de la malla de un modelo humano (borrando el resto de los puntos del modelo).

Después se fijaron las proporciones de ambos modelos. Se deseaba que el humano fuera bastante más bajito, así que el zombi le saca más de una cabeza. También se aumentó bastante el tamaño de la cabeza y de las manos originales del esqueleto –a fin de que no resultase un modelo demasiado serio– y finalmente se le añadieron unos ojos Nota: parece increíble cómo puede cambiar un esqueleto al ponerle ojos. Con las cuencas oculares vacías, el modelo da miedo mientras que con los ojos, el modelo pierde bastante horror y puede, en algunas posturas, hasta dar risa. Este efecto de caricatura puede aumentar si quitamos algunos huesos y damos mayor tamaño a algunas articulaciones como la cabeza, los pies, las manos, etc. En nuestro caso, estos cambios sólo fueron moderados.

Armas para la contienda

El siguiente paso fue preparar las armas para ambos contendientes. En el fichero base.obj, el lector podrá examinar las proporciones finales de los dos guerreros y sus armas. Originalmente el humano era un lancero y de hecho se preparó una lanza para él, pero finalmente no se preparó ninguna postura con ella. En lugar de esto se decidió preparar una espada de aspecto bastante bestial y un escudo. Esto se hizo en parte para dar un aspecto más combativo al personaje y en parte porque resultaba mucho más sencillo preparar posturas para él poniéndole un objeto en cada mano que haciéndole asir una lanza con las dos manos (probadlo y veréis). El diseño de estas armas no presentará ningún misterio para los lectores que hayan leído las notas que hemos ido publicando sobre «Imagine». El escudo son dos discos extruidos, uno de los cuales fue recortado. La hoja de la lanza se creó dibujando una forma bidimensional que fue extruida, editándose después sus puntos. Por último, la hoja de la espada se creó edi-

tando los puntos de un plano extruido (con práctica este método es más rápido de lo que parece a primera vista).

Una vez creadas las armas, estas fueron trasladadas y rotadas hasta colocarlas en los puños de nuestros audaces contendientes, después de lo cual fueron "atadas" a los objetos-mano como objetos-hijos.

La conversión a «POV»

Como los modelos iban a ser renderizados desde «POV», era preciso que éstos se almacenaran en una posición y escala que facilitará su manejo desde este programa. Como ocurría con «3D Studio» y «POV», «Imagine» tiene los ejes cambiados con respecto a nuestro trazador favorito, y fue preciso realizar unas cuantas operaciones sencillas para no hacer posteriores ajustes desde «POV». Hecho esto los modelos, una vez importados por «POV», quedaron con los pies sobre el suelo formado por el plano X-Z, a la altura Y=0. La cara de los modelos quedó mirando paralelamente al eje Z, en dirección al lado negativo del eje. Veamos cómo conseguir esto. Hay que seguir estos pasos:

1) Hay que rotar el modelo hasta que éste quede como puede verse en la foto: de pie y mirando hacia el espectador en la ventana Top, de modo que parezca visto desde arriba –y con los pies apuntando hacia el marco inferior de la ventana– desde arriba en la ventana Front. También deberá quedar extendido horizontalmente, con la cabeza hacia el lado derecho y mirando hacia abajo, en la ventana Right.

2) Después hay que situar los pies del personaje sobre el que será el suelo en «POV». Para ello iremos al menú "Display" y pulsaremos sobre la opción "Coordinates". Entonces aparecerán sobre la fila superior tres valores numéricos que irán variando conforme desplacemos el cursor. Estos valores corresponderán a la posiciones X,Y,Z sobre las que vaya pasando el cursor, al desplazarse por las ventanas de trabajo. Elegiremos una vista, colocaremos el cursor en la mismísima suela del zapato del personaje y tomaremos nota del valor devuelto en el eje Y (el valor del centro). Hecho esto sólo nos restará efectuar una traslación del modelo. Podemos hacerla manualmente pero resulta más preciso recurrir a la ventana de transformaciones (Alt-T). Escribiremos en la columna del eje Y el valor leído antes, pero dándole el signo opuesto (y no olvidéis el Return). Además deberemos pulsar sobre el flag de "World" antes de pinchar sobre "Perform".

3) Finalmente escogeremos un punto del personaje como centro en el plano X-Z. Lo ideal es hacer esto desde la ventana donde vemos al personaje desde arriba. Como antes, tomaremos nota de los valores devueltos por el cursor para este punto central (pero ahora para X y Z). Luego invocaremos nuevamente al gestor de transformaciones para el modelo y daremos los valores leídos (que son, claro está, la distancia a <0,0> de los ejes X y Z) cambiando el signo. Así, después de pinchar en "World" y "Perform", el muñeco estará perfectamente centrado en la posición desde la que debe ser "traducido" a «POV».

Una vez obtenidas la orientación y la posición con que el personaje aparecería en «POV» después de la traducción, llegó el momento de ponerse a preparar posturitas. Para nuestra batalla se necesitaban muchas posturas diferentes de humanos y zombis dando y recibiendo leña. Cada postura implicaba un fichero que sería traducido al formato de «POV» más tarde. Como las posturas se crearon partiendo de la orientación y posición ya logradas en el primer modelo grabado, todos los modelos-postura aparecieron en el universo de «POV» con los pies sobre la posición <0,0,0>. Esto ya garantizaba que no iba a costar mucho ir situándolos en el campo de batalla. Después de llevar a cabo los tres pasos ya descritos se grabó el modelo y se realizaron en él los cambios necesarios para cada postura. Cada vez que se obtenía una nueva postura, se la grababa y se reseteaba al Editor de Detalles para leer de nuevo la postura original sobre la que se deseaba trabajar para obtener todas las demás. Por supuesto, también se podía emplear alguna de las posturas ya obtenidas, cuando ello resultaba más fácil.

Otra vez el 3dto3d

Supongamos que ya tenemos todos los ficheros-postura. Procederemos a convertir cada obj con una línea como...

3dto3d postura.obj /i2 /o2 /s45

Esto creará un fichero inc que será directamente digerible por «POV». Hay, sin embargo, un problema. La conversión a «POV» creará un declare para cada objeto dentro de un fichero .inc que puede tener varios Megas de tamaño. No se crea, sin embargo, ningún #declare para la unión de todos los objetos constituyentes del modelo. Y tampoco se añaden texturas, ni pigmentos, ni luces, ni cámaras. (Nota: nosotros no hemos hallado ninguna opción de 3dto3d para ello).

Esto derivó en una terrible pesadilla pues obligó a emplear un Editor de Textos en cada uno de los 25 archivos-postura para colocar manualmente las texturas y eliminar los declares (el lector podrá imaginarlo si tiene en cuenta que cada .inc obtenido tiene un tamaño que oscila entre los 3.5 y los 4 Megas para cada modelo-postura).

Desde los ficheros escénicos de «POV» cada postura se trata así:

object{#include "posturaX.inc" rotate <x,y,z> translate <nx,ny,nz> }

Naturalmente la postura podría haberse grabado como un único objeto antes de realizar la traducción pero ello hubiera implicado que sólo habría un color por postura. Finalmente hay que recordar que 3dto3d graba también un fichero udo por cada postura. En nuestro caso este fichero no nos interesa demasiado, pero puede ser conveniente que toméis nota de que, al principio del mismo, se añaden las dimensiones del modelo convertido en los distintos ejes. Esto puede ser muy útil para estimar las distancias en la colocación de los modelos.

Nota: 3dto3d solamente convierte ficheros .obj de «Imagine», no objetos creados desde el Editor de Formas. Para traducir objetos creados desde este módulo será preciso grabarlos desde el Editor de Detalles y darles la terminación .obj.

Las posturas

Para la batalla se prepararon 25 posturas del zombi y del humano. Hay golpes, caídas, cadáveres e incluso posturas de los personajes volando (tras recibir supuestamente un cañonazo). Pero lo cierto es que hubieran sido necesarias muchas más para preparar una batallita en condiciones. Desafortunadamente cada postura, humana o zombi, demanda mucha memoria de «POV» (¡unos 8 Megas!). En estas circunstancias, ¿para qué hacer más? Aquí se echó en falta una buena utilidad de reducción de polígonos. Con ella hubiera sido sencillo reducir el tamaño de los archivos a dimensiones manejables.

Podéis encontrar los ficheros obj con los modelos en el CD-ROM.

Trazado de batallas

En portada podéis admirar diversos modelos realizados por los lectores como apoyo a la idea de un enfrentamiento virtual entre humanos y orcos. Esta idea inicial ha sufrido ciertas modificaciones debidas a limitaciones en el tiempo y en el hardware disponibles. Seguidamente estudiaremos estos problemas teniendo en mente futuras colaboraciones de los rendermaníacos en próximas portadas.

Sin duda, la portada del presente número supondrá una sorpresa, y probablemente una decepción. para todos aquellos que esperasen encontrar una imagen repleta de orcos y humanos luchando. Las razones de la no comparecencia del bando orco ya han sido explicadas, por lo que no vamos a insistir en ellas. En lugar de esto explicaremos en detalle todos los problemas que se dieron en la generación de esta escena, a fin de prevenir su repetición en próximas ocasiones.

Falta de memoria

Las limitaciones antes mencionadas han determinado en gran medida las modificaciones hechas a la idea original. El problema más grave ha sido, por supuesto, el de la memoria: la portada había de generarse en un PC con 64 Megas de RAM y cada personaje, humano o zombi, precisaba de 8. Esto ha limitado enormemente el número previsto de contendientes y constituye un monumental fallo de planificación por nuestra parte, ya que los requerimientos de memoria del guerrero humano eran conocidos desde hacia tiempo. (Nota del autor: en mi descargo debo añadir que, aunque conocía el problema podía hacer bien poco por remediarlo puesto que no dispongo de ningún reductor de polígonos. Además resulta bastante difícil crear buenos modelos con cientos –en vez de miles– de caras. En estos casos suele ser preciso realizar bitmaps para disimular la falta de detalle de estos modelos y, como ya hemos comentado antes, no somos especialmente hábiles como grafista 2D).

Los modelos de los rendermaníacos

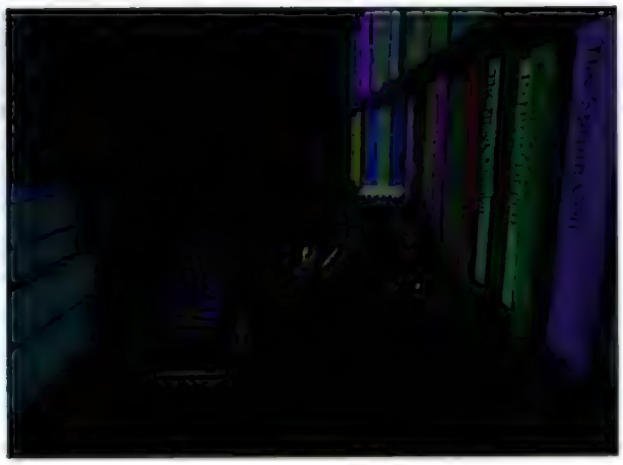
Para la portada hemos seleccionado las máquinas de guerra enviadas por Óscar Álvarez Díaz, Daniel Susín Acebo, Julián Hierro García y Carlos Vilas Arias. También estaba previsto incluir el maravilloso mago de Juan Carlos Jiménez Méndez, pero al final no fue incluido debido a la falta de RAM (estaba previsto darle el mando del ejército zombi pero al final, en vista del escaso número de no-muertos que comparecieron...). También hubo otros modelos que, por una u otra razón, no pudieron ser incluidos en esta lista. Entre ellos podemos citar a Roberto Reboiro Jato. Su catapulta merecía ser incluida aquí pero, desafortunadamente, Roberto sólo envió los mdl. Nosotros no teníamos instalado «Moray» y, debido a la falta de tiempo, no fue posible efectuar la conversión.

Por otro lado la preparación de los modelos incluidos se llevó su tiempo. Lógicamente, al no haberse publicado instrucciones sobre este particular, cada lector envió su trabajo con una escala y orientación de ejes distinta. Esto obligó a leer los fuentes y a realizar ligeros cambios para adaptar los modelos a la escala y orientación pensados para la portada. Esta tarea fue bastante fastidiosa ya que nadie pensó en poner comentarios en sus fuentes indicando las dimensiones de sus modelos y la altura a la que se había colocado el suelo. Además, casi ningún lector pensó en centrar los modelos en los ejes, lo cual hubiera facilitado mucho la tarea de situarlos en la escena.

Otro detalle fastidioso en que cayeron bastantes lectores fue el de renombrar las declaraciones de bastantes texturas de madera o metal. Como el fichero de generación de la portada incluía varias catapultas que empleaban las mismas texturas de madera, esto ocasionó que se produjeran cambios en el acabado de otros modelos que empleaban las mismas texturas y también aparecieron algunos bugs que requirieron un fastidioso proceso de depuración. Finalmente, algún lector ha incluido modelos cuyos fuentes demandaban algún bitmap que no ha sido proporcionado.

La portada final

La idea inicial para la portada contemplaba el uso de las viejas librerías de pueblos y castillos medievales que se publicaron en los primeros números del suplemento Rendermanía. Sin embargo, había varios factores que desaconsejaban el uso de estas librerías para construir los fondos de la portada. El problema principal venía dado por los requerimientos de memoria. La razón de este excesivo gasto radica en que las dos librerías están agrupadas en dos



grandes ficheros que emplean un gran número de declaraciones (para una próxima versión esto cambiará y la librería se descompondrá en un buen número de ficheros inc más pequeños). Por otro lado, además, no deseábamos emplear nuevamente estas librerías en una portada en tanto no se incluyeran mejoras como nuevos edificios, nuevas texturas, etc.

Entonces, después de quemar neuronas durante algún tiempo, apareció la siguiente idea: ¿por qué no emplear a los guerreros y al resto de los modelos como si fuesen miniaturas que estuviesen librando una batalla de juego al estilo de las de «Warhammer»? Para crear esta impresión únicamente era necesario situar los modelos sobre una mesa y colocar también los objetos típicos que pueden verse en las mesas; lápices, teclados de ordenador, libros, etc. Por supuesto de ahí a la utilización de los objetos-libros de Sonya Roberts sólo había un paso. Los libros podían ser utilizados como plataformas por las "miniaturas" y además aportaban un buen contraste.

El siguiente paso, una vez colocadas las filas de libros, fue decidir la orientación de la cámara. Como no se deseaba perder más tiempo añadiendo detalles adicionales como paredes, muebles, etc., se enfocó la cámara de modo que las filas de libros actuasen como fondo. Al mismo tiempo, como se precisaba enfocar un buen trozo de mesa donde colocar las miniaturas, la cámara se dispuso de tal manera que pudiese captar tres zonas diferenciadas: la librería de fondo, las miniaturas de la mesa y las miniaturas que luchan en primer plano sobre una fila de libros.

Indudablemente se echan a faltar muchos detalles en la escena: mesa debería tener una lámpara y no bieran venido mal algunos elementos más (de los típicos que pueden hallarse en cualquier mesa). También hubiera quedado bonito poner a los pies de cada modelo una peana, como las que suelen tener las miniaturas, pero no se disponía del tiempo suficiente para crear estos detalles. También es una pena no haber podido meter varias parejas más de combatientes pero...

Reglas para posibles portadas futuras

No renunciamos a crear una auténtica batalla entre orcos y an gún día, si se resuelven estos problemas. Mientras tanto confiamos en que los autores de los modelos empleados no se hayan molestado por los cambios hechos a la idea original (¡no hubo más remedio!) Si algún día la batalla puede celebrarse, estos modelos serán utilizados.

Ahora veamos las cláusulas que regirán la creación de las próximas portadas hechas con ayuda de los lectores:

1) Cada 4 meses se publicará una portada que incorporará modelos hechos por los lectores. El motivo de la portada deberá ceñirse –aunque sea tangencialmente como en el presente caso– a un tema anunciado previamente. Si la respuesta de los lectores es insuficiente, la portada se cancelará.

2) Para cada portada se publicarán reglas especiales adicionales, según sea el caso. Estas reglas se referirán a las dimensiones que deberán tener los modelos y también a su orientación y al programa en que se realizarán las portadas (normalmente «POV»).

3) Los fuentes de generación de los modelos enviados (en el caso de «POV») deberán indicar comentarios sobre la orientación del morro del modelo y sobre sus dimensiones. Todos los modelos deberán estar siempre reposando el plano X-Z, con los pies a la altu e Y=0 (y preferentemente mirando en la dirección -Z).

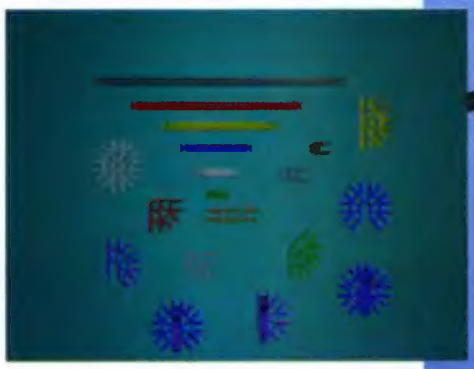
4) Cada metro virtual equivaldrá a 10 unidades.

El tema elegido para la próxima portada popular será "Mechs". Si utilizáis «3D Studio» o «Imagine» se agradecerá que establezcáis las jerarquías de objetos a que, si lo preferís, podéis enviar las "posturas" que creáis adecuadas. En caso de que se reciba un número de modelos muy alto, serán empleados los que se consideren mejores o más apropiados para el motivo de la portada. Nota: esta portada no tiene por qué ser una escena de guerra. También podéis enviar sugerencias acerca de cómo imagináis vosotros la portada o de cómo preferís que se empleen vuestros modelos en ella.

Nota importante. Podéis remitirnos vuestros trabajos o consultas, bien por carta a la dirección que figura en la segunda página de Pcmanía, o vía e-mail a rendermania.pcmania@hobbypress.es

Babel Virtual

Hoy nuestro foro es un verdadero Babel en cuanto a las técnicas y programas empleados por los autores. 3D Studio, Imagine, POV, Moray e incluso MAX han sido utilizados para crear los trabajos que podéis admirar hoy. Extraterrestres, soldados espaciales, máquinas medievales, Mechs... Las herramientas, los temas, las preferencias y las ideas son diferentes en cada caso pero hay algo que todos los rendermaníacos tienen en común: la pasión por la infografía.



A **José Antonio Siñuela Rajo** y a su hijo **David** les gustó la original idea de Jorge Martín Cuervo, el cual publicó en el número 6 de Rendermanía una librería virtual de piezas del juego Tente. Por ello, José Antonio y David han retomado la idea y han creado una librería de piezas del juego K´nex. Las piezas, creadas con 3D Studio por José Antonio, fueron empleadas por David (de 12 años) para montar el helicóptero de juguete que padre e hijo nos han enviado como ejemplo. Aquí se me ocurre una pequeña idea: alguien podría diseñar un juego de construcción partiendo de cero y enviar modelos construidos con él. ¿Quién sabe? Quizá, si la idea está bien pensada, su autor podría registrarla. Y aquí, si la idea atrae a un número lo suficientemente alto de lectores, podríamos dedicar una portada al tema "Juegos de Construcción". Os felicito por vuestro trabajo.

El Foro del Lector

Roberto Celorrio del Pino nos envía un modelo basado en los marines espaciales del capítulo de los ángeles Sangrientos, del juego Warhammer 40000. Este fantástico modelo está, como podéis ver por las fotos, totalmente articulado, y Roberto nos adjunta no sólo la malla correspondiente, sino también las texturas. En su carta, Roberto hace una autocrítica bastante dura, ya que el modelo no es exactamente igual a la miniatura en que está basado, pero en mi opinión el único defecto reseñable está –como también señala el propio autor– en la hombrera, ya que ésta secciona el antebrazo. El modelo ha sido creado con 3D Studio haciendo uso del comando Fit y aplicando operaciones Csg y de deformación. Roberto ha declarado su modelo como de libre uso y tan sólo pide que se mencione su nombre. ¡Enhorabuena por tu magnífico trabajo! Espero ver luchar algun día a tu infante en algún ¡Waaagh! o en alguna invasión Geneleaster.

Ángel Ruiz Guijarro nos ha enviado su primer trabajo con Imagine. Se trata de un modelo femenino creado para hacer compañía a Ray y defenderle de los abusos del grandullón que apareció en el número 5 de Rendermanía. Pues bien, amigo Ángel, siento desilusionarte, pero Ray ya está enamorado desde hace tiempo. Su corazón virtual suspira por los huesos de una de las chicas de Tomwoof. Ella, sin embargo, le ha dado calabazas hasta ahora aduciendo que no es más que un muñeco de madera. En fin....

En cuanto a lo que dices sobre los pechos, podrías emplear la edición de puntos, el magnetismo o el Editor de Formas o una mezcla de técnicas. No existen reglas fijas para el infografista. Cada uno modela empleando las herramientas que prefiere. De todas formas te hago notar que unos pechos "más reales" desentonarían en tu muñeco (yo se los quitaría y le pondría mejor un lanzacohetes). En cuanto al traspaso de los modelos desde Imagine a POV, el único conversor que conozco que trata los ficheros .obj es 3dto3d de Thomas Baier, que se incluye en este número. Ah, en cuanto a lo de trastear en tu modelo, yo prefiero casi siempre empezar desde cero.

Benjamin Albares Moreno regresa a estas páginas para enviar algunas imágenes que representan un gran avance con respecto a sus anteriores trabajos. Benyi nos envía una extensa carta en la que comenta el proceso de desarrollo de sus escenas y en la que pide unos comentarios. Pues bien, tus escenas me han parecido magníficas pero... ¿por qué tienen tan poco color? El efecto general es un poco frío. Quizá habría sido conveniente dar un color distinto a la cabeza del primer plano de Futuras, para que resaltara más sobre el fondo, o bien cambiar el color de este. Por otro lado, estoy de acuerdo contigo en el papel que juegan las texturas metálicas en las robots, y me ha impresionado el estudio que has hecho para el Escorpión y el modelo en sí, aunque creo que hubiera quedado mejor como escorpión que como nave espacial. ¡Hubiera dado mucho más asco! También habrías debido poner un fondo menos pobre en la escena del escorpión (a fin de cuentas echaste mucho esfuerzo en el modelo, ¿Por qué no entonces algo más de trabajo para presentarlo?). En fin, como modelador has mejorado un montón y estoy deseando ver acabada a la chica que estás preparando con Metaballs.

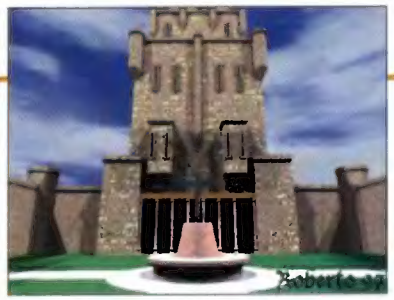
Alberto Riera Sánchez nos ha enviado las que quizá serán sus últimas imágenes con 3D Studio, ya que piensa pasarse a MAX. Debido a un desastre en su disco duro, Alberto no ha enviado todos los ficheros de generación pero su carta (que debería haber enviado también como fichero) me ha convencido de la autoría de las imágenes. Alberto me ha pedido una pequeña crítica así que ahí va: tu punto fuerte parece ser la ambientación de las escenas más que el modelado en sí (contrariamente a como sucede con la mayoría de los rendermaníacos). Pandemonium2, por ejemplo, resulta asombrosamente tétrica, a pesar de la excesiva sencillez del templo (y de la textura que le has puesto). Algo parecido ocurre con las demás escenas: en Alien, el modelo (hecho con metaballs) no parece demasiado resultón en sí mismo, pero en el contexto de la imagen transmite una sensación de peligro (por cierto, creo que acertaste al no ponerle ojos). También me ha gustado la escena del Androide. En resumen: creo que si potencias la complejidad de tus modelos conseguirás unas escenas espléndidas. Todo lo demás: la composición, las ideas, la atmósfera, etc., me ha gustado bastante.

Ming Lee Chuei ha enviado una escena donde puede verse un modelo fiel del fantástico Marauder, cuyo diseño sirvió de base para el Mech que incluí como ejemplo en el número 7 de Rendermanía. Tengo que confesar que tu Mech me parece mucho más atractivo que el mío, aunque espero superarme próximamente si se llega a celebrar una batalla entre Mechs en alguna de las próximas portadas.

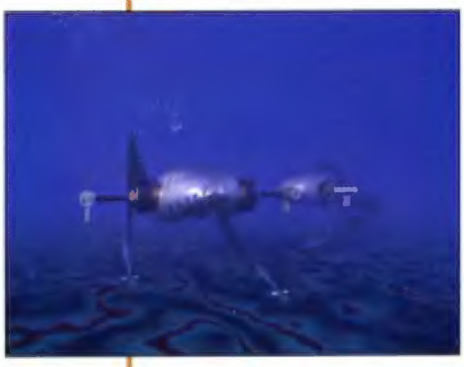
Sin embargo, tengo que decir que uno de tus discos llegó defectuoso. ¿Contenía las mallas de este modelo? En ese caso espero que me las envíes de nuevo junto con alguna nota indicando algo sobre el proceso de creación. En cuanto a tu pregunta sobre cómo enviarnos animaciones, supongo que lo ideal sería hacerlo en un CD, ya que todo el mundo tiene uno (yo por ejemplo no dispongo aún de ningún removible).

Carlos Vilas Arias ha decidido apoyar al bando humano con varias catapultas hechas con Moray y POV. En principio su plan consistía en enviar un mago-enano, pero al final se decidió por los lanza-pedruscos, de los cuales uno es más bien un transporte de munición que una máquina de guerra. ¡Gracias por tu apoyo, Carlos! Y la próxima vez no empleés tanta niebla en la escena.

El Foro del Lector



Roberto de los Ojos Gracia nos envía sus primeras escenas con POV. En su carta Roberto nos cuenta cómo ha pagado su condición de Povnovato. La verdad es que me reí bastante con lo que mencionas sobre las ventanas del rascacielos (Roberto las puso todas a mano y solo después se enteró de la existencia de la sentencia While). En fin, a mí me ha ocurrido algo parecido –si no peor– con mis modelos y 3dto3d. En cuanto a While, me parece que aún no has acabado de pillarle el tranquillo a las sentencias de programación, ya que dices que no supiste utilizar esta sentencia con las ventanas de la escena de la residencia. Por otro lado tus imágenes me han parecido magníficas para ser las primeras y espero ver pronto las siguientes.

Francisco Palao Reinés es un nuevo aficionado a la infografía que nos envía sus primeros trabajos con MAX. Se trata de dos nuevas máquinas de guerra diseñadas para participar en la batalla virtual. El problema es que no se aún nada de MAX y no pude, por tanto, editar los modelos para incorporarlos a la batalla (además, no conozco ningún traductor para MAX). Sorry. En cuanto a la crítica solicitada, creo que aún puede mejorar mucho en todo lo relativo al modelado aunque, como no conozco MAX, no puedo dar demasiados detalles útiles. Sorry otra vez.

Tengo que agradecer a **Víctor Tovío Ciércoles** el envío de 3dto3d. Lo cierto es que hasta ahora me había pasado inadvertida esta estupenda herramienta de conversión de formatos. En cuanto hice las primeras pruebas con él lamenté no haber tenido antes este programa. ¡La de trabajo inútil que me hubiera ahorrado sólo en este número! Víctor nos envía hoy sus primeros trabajos: un par de escenas y una animación de un Mech modelado con Imagine y exportado a POV. Víctor ha incluido en su carta una descripción completa del proceso de traducción del modelo a POV (lo cual fue lo primero que llamó mi atención). Tanto las escenas como la animación se han renderizado desde POV y Víctor solicita otra de esas "críticas constructivas" que hacen que tanta gente no duerma bien por las noches. Pues bien, básicamente lo que puedo decirte es que aún puedes mejorar mucho en el apartado de modelado. Dedica por ahora más tiempo a esto antes que a otras cosas como la ambientación o la animación. Gracias de nuevo por el 3dto3d. Espero tus próximos trabajos.